

This Page Is Inserted by IFW Operations
and is not a part of the Official Record

BEST AVAILABLE IMAGES

Defective images within this document are accurate representations of the original documents submitted by the applicant.

Defects in the images may include (but are not limited to):

- BLACK BORDERS
- TEXT CUT OFF AT TOP, BOTTOM OR SIDES
- FADED TEXT
- ILLEGIBLE TEXT
- SKEWED/SLANTED IMAGES
- COLORED PHOTOS
- BLACK OR VERY BLACK AND WHITE DARK PHOTOS
- GRAY SCALE DOCUMENTS

IMAGES ARE BEST AVAILABLE COPY.

As rescanning documents *will not* correct images,
please do not report the images to the
Image Problem Mailbox.

PATENT ABSTRACTS OF JAPAN

(11)Publication number : 2000-286709

(43)Date of publication of application : 13.10.2000

Jc929 U.S. PTO
10/073397

(51)Int.Cl.

H03M 7/14
G11B 20/14

(21)Application number : 2000-020171

(71)Applicant : VICTOR CO OF JAPAN LTD

(22)Date of filing : 28.01.2000

(72)Inventor : HAYAMIZU ATSUSHI

(30)Priority

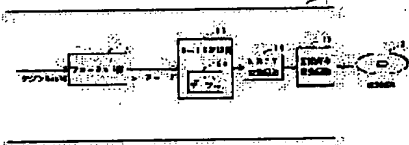
Priority number : 11023316 Priority date : 29.01.1999 Priority country : JP

(54) MODULATION METHOD, MODULATION DEVICE, DEMODULATION METHOD, DEMODULATION DEVICE AND RECORDING MEDIUM

(57)Abstract:

PROBLEM TO BE SOLVED: To improve an encoding rate by permitting a code table to store a code word corresponding to an input data word and state information showing an encoding table used for encoding the next input data word and selecting the plural encoding tables.

SOLUTION: An 8-15 modulator 13 converts the input data word of eight bits, which is inputted through a format part 12, into the code word of 15 bits by referring to a code table 120. An NRZI conversion circuit 14 receives a code word, generates an NRZI signal and records it in a recording medium 2 through a recording medium driving circuit 15. At the time of constituting the code table 120, the code word is allocated based on a limit that a minimum run length is 3T and a maximum run length is 11T with a channel bit as T. The table is divided into seven groups 7 based on a transition state and the group can be selected in accordance with a following input data word. Thus, the run length limit is satisfied even if the code word is directly connected.



LEGAL STATUS

[Date of request for examination]

[Date of sending the examiner's decision of rejection]

[Kind of final disposal of application other than the examiner's decision of rejection or application converted registration]

[Date of final disposal for application]

[Patent number]

[Date of registration]

[Number of appeal against examiner's decision of rejection]

[Date of requesting appeal against examiner's decision of rejection]

[Date of extinction of right]

Copyright (C); 1998,2000 Japan Patent Office

(19) 日本国特許庁 (JP)

(12) 公開特許公報 (A)

(11) 特許出願公開番号
特開2000-286709
(P2000-286709A)

(43) 公開日 平成12年10月13日 (2000. 10. 13)

(51) Int.Cl. ⁷	識別記号	FI	テーマト* (参考)
H03M 7/14		H03M 7/14	B
G11B 20/14	341	G11B 20/14	341A

審査請求 未請求 請求項の数11 OL (全 23 頁)

(21) 出願番号 特願2000-20171(P2000-20171)
 (22) 出願日 平成12年1月28日 (2000. 1. 28)
 (31) 優先権主張番号 特願平11-23316
 (32) 優先日 平成11年1月29日 (1999. 1. 29)
 (33) 優先権主張国 日本 (JP)

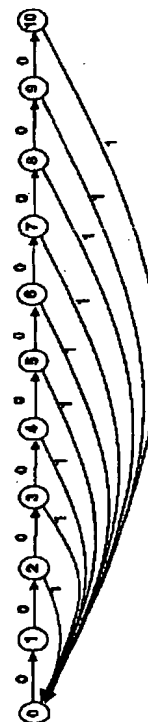
(71) 出願人 00004329
 日本ビクター株式会社
 神奈川県横浜市神奈川区守屋町3丁目12番
 地
 (72) 発明者 速水 淳
 神奈川県横浜市神奈川区守屋町3丁目12番
 地 日本ビクター株式会社内
 (74) 代理人 100093067
 弁理士 二瓶 正敬

(54) 【発明の名称】 変調方法、変調装置、復調方法、復調装置及び記録媒体

(57) 【要約】

【課題】 8ビットデータを16ビット符号に変調する
 EFM+方式より更にコード化レートを向上させる。

【解決手段】 入力データ語を符号語に符号化するための
 符号化テーブル1として入力データ語に対して複数の
 符号化テーブルを用い、前記複数の符号化テーブルは入
 力データ語に対応する符号語と、次の入力データ語を符
 号化するための符号化テーブルを選択するための状態情
 報を有するとともに、所定の入力データ語に対する特定
 の符号化テーブルにおける符号語と他の特定の符号化テ
 ーブルにおける符号語をそれぞれNRZI変調した信号
 が逆極性(「1」の数の偶奇性が異なる)である。



【特許請求の範囲】

【請求項1】 p ビットの入力データ語を所定のランレングス制限規則を満たす q (但し、 $p < q$) ビットの符号語に符号化すると共に、前後の符号語を直接結合しても前記所定のランレングス制限規則を満たすように符号化する変調方法であって、

入力データ語を符号語に符号化するために複数の符号化テーブルを用いると共に、前記各符号化テーブルは、それぞれの入力データ語に対応して、符号語と、この符号語に直接結合しても前記ランレングス制限規則を満たすような次の符号語を得るために次の入力データ語を符号化するのに使用する前記符号化テーブルを示す状態情報とを格納し、さらに前記複数の符号化テーブルのうちの特定の符号化テーブルと他の特定の符号化テーブルは、所定の入力データ語に対しては、入力データ語に対応して格納されているそれぞれの符号語をNRZI変換した信号が逆極性となるように符号語が割当てられており、前記所定の入力データ語を符号化する際に前記特定の符号化テーブルが指定されている場合には、前記特定の符号化テーブルと前記他の特定の符号化テーブルとを適宜選択することにより、DSV制御を行いながら符号化するようにしたことを特徴とする変調方法。

【請求項2】 $p = 8$ 、 $q = 15$ であって、前記ランレングス制限規則は、符号語をNRZI変換した信号の最小ランレングスが3T (但し、Tは符号語のチャネルビット周期)、最大ランレングスが11Tであることを特徴とする請求項1記載の変調方法。

【請求項3】 p ビットの入力データ語を所定のランレングス制限規則を満たす q (但し、 $p < q$) ビットの符号語に符号化すると共に、前後の符号語を直接結合しても前記所定のランレングス制限規則を満たすように符号化する変調装置であって、

入力データ語を符号語に符号化するための符号化テーブルを複数備え、

前記各符号化テーブルは、それぞれの入力データ語に対応して、符号語と、この符号語に直接結合しても前記ランレングス制限規則を満たすような次の符号語を得るために次の入力データ語を符号化するのに使用する前記符号化テーブルを示す状態情報とを格納しており、さらに前記複数の符号化テーブルのうちの特定の符号化テーブルと他の特定の符号化テーブルは、所定の入力データ語に対しては、入力データ語に対応して格納されているそれぞれの符号語をNRZI変換した信号が逆極性となるように符号語が割当てられた構成となっていることを特徴とする変調装置。

【請求項4】 p ビットの入力データ語を所定のランレングス制限規則を満たす q (但し、 $p < q$) ビットの符号語に符号化すると共に、前後の符号語を直接結合しても前記所定のランレングス制限規則を満たすように符号化する変調装置であって、

入力データ語を符号語に符号化するための符号化テーブルを複数備え、

前記各符号化テーブルは、それぞれの入力データ語に対応して、符号語と、この符号語に直接結合しても前記ランレングス制限規則を満たすような次の符号語を得るために次の入力データ語を符号化するのに使用する前記符号化テーブルを示す状態情報とを格納しており、

さらに前記複数の符号化テーブルのうちの特定の符号化テーブルと他の特定の符号化テーブルは、所定の入力データ語に対しては、入力データ語に対応して格納されているそれぞれの符号語をNRZI変換した信号が逆極性となるように符号語が割当てられた構成となっており、前記所定の入力データ語を符号化する際に前記特定の符号化テーブルが指定されている場合には、前記特定の符号化テーブルと前記他の特定の符号化テーブルとをDSV制御を行うように適宜選択して符号化する選択手段を備えたことを特徴とする変調装置。

【請求項5】 p ビットの入力データ語を所定のランレングス制限規則を満たす q (但し、 $p < q$) ビットの符号語に符号化すると共に、前後の符号語を直接結合しても前記所定のランレングス制限規則を満たすように符号化する変調装置であって、

入力データ語を符号語に符号化するための符号化テーブルを複数備え、

前記各符号化テーブルは、それぞれの入力データ語に対応して、符号語と、この符号語に直接結合しても前記ランレングス制限規則を満たすような次の符号語を得るために次の入力データ語を符号化するのに使用する前記符号化テーブルを示す状態情報とを格納しており、

さらに前記複数の符号化テーブルのうちの特定の符号化テーブルと他の特定の符号化テーブルは、所定の入力データ語に対しては、入力データ語に対応して格納されているそれぞれの符号語をNRZI変換した信号が逆極性となるように符号語が割当てられた構成となっており、前の入力データ語を符号化する際に読み出された状態情報と入力データ語とが供給され、入力データ語が前記所定の入力データ語であって、かつ状態情報が前記特定の符号化テーブルを示すか否かを判定する判定手段と、前記判定手段の判定結果に基づいて、複数又は1つの符号化テーブルから個々のバス毎に入力データ語に対応する符号語を読み出す読み出し手段と、

前記バス毎に前記符号化テーブルから読み出された符号語を記憶する複数のバスメモリと、

前記複数のバスメモリにそれぞれ記憶されている符号語をNRZI変換した信号のDSVの総和を記憶する複数のDSVメモリと、

前記DSVの総和の絶対値が小さい方のバスメモリに記憶されている符号語を符号化データとして選択する選択手段とを、

有することを特徴とする変調装置。

【請求項6】 pビットの入力データ語が所定のランレングス制限規則を満たすq（但し、 $p < q$ ）ビットの符号語に符号化されていると共に、前後の符号語を直接結合しても前記所定のランレングス制限規則を満たすように符号化されているqビットの符号語をpビットの入力データ語に復調する復調方法であって、
 入力される符号語を復号するために入力データ語を符号化した際に用いられた複数のコードテーブルを用いると共に、前記各コードテーブルは、それぞれの入力データ語に対応して、符号語と、この符号語に直接結合しても前記ランレングス制限規則を満たすような次の符号語を得るために次の入力データ語を符号化するのに使用する前記コードテーブルを示す状態情報とを格納し、さらに前記複数のコードテーブルのうちの特定のコードテーブルと他の特定のコードテーブルは、所定の入力データ語に対しては、入力データ語に対応して格納されているそれぞれの符号語をNRZI変換した信号が逆極性となるように符号語が割当てられており、
 前の符号語のLSB側のゼロラン長に基づいて後の符号語の取り得る状態番号を検出すると共に、後の符号語を符号化した際に使用されたコードテーブルを示す状態番号を算出し、検出された状態番号と算出された状態番号と前の符号語とから前の符号語を復調することを特徴とする復調方法。

【請求項7】 pビットの入力データ語が所定のランレングス制限規則を満たすq（但し、 $p < q$ ）ビットの符号語に符号化されていると共に、前後の符号語を直接結合しても前記所定のランレングス制限規則を満たすように符号化されているqビットの符号語をpビットの入力データ語に復調する復調装置であって、
 入力される符号語を復号するために入力データ語を符号化した際に用いられた複数のコードテーブルを備え、前記各コードテーブルは、それぞれの入力データ語に対応して、符号語と、この符号語に直接結合しても前記ランレングス制限規則を満たすような次の符号語を得るために次の入力データ語を符号化するのに使用する前記コードテーブルを示す状態情報とを格納し、さらに前記複数のコードテーブルのうちの特定のコードテーブルと他の特定のコードテーブルは、所定の入力データ語に対しては、入力データ語に対応して格納されているそれぞれの符号語をNRZI変換した信号が逆極性となるように符号語が割当てられており、
 前の符号語のLSB側のゼロラン長に基づいて、後の符号語の取り得る状態番号を検出する検出手段と、
 後の符号語を符号化した際に使用されたコードテーブルを示す状態番号を算出する算出手段と、
 前記検出手段からの状態番号と前記算出手段からの状態番号と前の符号語とを用いて前記複数のコードテーブルを参照し、前の符号語を復調する復調手段と、
 を備えたことを特徴とする復調装置。

【請求項8】 pビットの入力データ語が所定のランレングス制限規則を満たすq（但し、 $p < q$ ）ビットの符号語に符号化されていると共に、前後の符号語を直接結合しても前記所定のランレングス制限規則を満たすように符号化されているqビットの符号語をpビットの入力データ語に復調する復調方法であって、
 後の符号語を符号化した際に使用されたコードテーブルを示す状態番号を算出して、前の符号語と後の符号語の状態番号に対応して復号データ語が蓄積されている復号テーブルを参照し、前の符号語を復調することを特徴とする復調方法。

【請求項9】 pビットの入力データ語が所定のランレングス制限規則を満たすq（但し、 $p < q$ ）ビットの符号語に符号化されていると共に、前後の符号語を直接結合しても前記所定のランレングス制限規則を満たすように符号化されているqビットの符号語をpビットの入力データ語に復調する復調装置であって、
 後の符号語を符号化した際に使用されたコードテーブルを示す状態番号を算出する算出手段と、
 前記算出手段からの状態番号と前の符号語とを用いて復号テーブルを参照し、前の符号語を復調する復調手段とを備え、
 前記復号テーブルは、前の符号語と後の符号語の状態番号に対応して復号データ語が蓄積されていることを特徴とする復調装置。

【請求項10】 pビットの入力データ語を所定のランレングス制限規則を満たすq（但し、 $p < q$ ）ビットの符号語に符号化すると共に、前後の符号語を直接結合しても前記所定のランレングス制限規則を満たすように符号化された符号語がNRZI変換されて記録されている記録媒体であって、
 入力データ語を符号語に符号化するために複数の符号化テーブルを用いられると共に、前記各符号化テーブルは、それぞれの入力データ語に対応して、符号語と、この符号語に直接結合しても前記ランレングス制限規則を満たすような次の符号語を得るために次の入力データ語を符号化するのに使用する前記符号化テーブルを示す状態情報とを格納し、さらに前記複数の符号化テーブルのうちの特定の符号化テーブルと他の特定の符号化テーブルは、所定の入力データ語に対しては、入力データ語に対応して格納されているそれぞれの符号語をNRZI変換した信号が逆極性となるように符号語が割当てられており、
 前記所定の入力データ語を符号化する際に前記特定の符号化テーブルが指定されている場合には、前記特定の符号化テーブルと前記他の特定の符号化テーブルとを適宜選択することにより、DSV制御を行いながら符号化された符号語がNRZI変換されて記録されていることを特徴とする記録媒体。

【請求項11】 $p = 8$ 、 $q = 15$ であって、前記ラン

レンジ制限規則は、符号語をNRZI変換した信号の最小ランレンジが3T（但し、Tは符号語のチャンネルビット周期）、最大ランレンジが11Tであることを特徴とする請求項10記載の記録媒体。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】本発明は、デジタル信号を光ディスク、磁気ディスクなどの記録媒体に記録再生したり、デジタル信号を伝送するのに好適な変調方法、変調装置、復調方法、復調装置及び記録媒体に関するものである。

【0002】

【従来の技術】一般に、光ディスクに記録されるビット長は、記録再生の光伝送特性や、ビット生成に係わる物理的な制約から最小ランレンジ（最小ビット又はランド長）の制限、クロック再生のしやすさから最大ランレンジ（最大ビット又はランド長）の制限、さらにはサーボ帯域などの保護のために、被記録信号の低域成分の抑圧特性を持つよう記録信号を変調して設けられる必要がある。この制限を満たす従来の変調方式のうち、最小ランレンジを3T（T=チャンネルビットの周期）、最大ランレンジを11Tとしたものに、CD（コンパクト・ディスク）に用いられているEFM（8-14変調）方式や、DVD（デジタル・バーサタイル・ディスク）に用いられているEFM+方式が知られている。

【0003】前者のEFM方式では、3ビットのマージンビットを介した際に、ランレンジ制限（RL）の規則RL（2，10）により「1」と「1」の間の「0」が2個以上かつ10個以下になるように8ビット符号を14チャンネルビットの符号に変換し、更に24チャンネルビットの同期信号と14チャンネルビットのサブコードを付加した後、前後の2つの符号を結合した場合にもRL規則（2，10）を満たすようにDSV（デジタル・サム・バリエーション）制御により最適なビットパターン3ビットのマージンビットを選択して、このマージンビットを介して14ビット符号間を連結して17ビット符号語に変換し、次いでこれをNRZI変換して記録する。

【0004】後者のEFM+方式では、例えば特開平8-31100号公報に示されるように上記のマージンビットを用いずに8ビットデータを直接、16ビット符号に符号化する。この方法では、符号化テーブルの16ビット符号は、1つの符号がRL規則（2，10）を満たし、かつ前後の2つの符号を結合した場合にもRL規則（2，10）を満たすように結合される。また、符号化テーブルの一部が2重化されており、この2重化部分は、対応する符号の組がお互いにDSVの変化量が正負逆でかつ絶対値が近くなるように構成されるとともに、DSVの変化量の絶対値が大きい符号が配置されている。したがって、この16ビット符号のEFM+方式

は、17ビット符号のEFM方式よりコード化レートが約6%向上する。

【0005】

【発明が解決しようとする課題】しかしながら、従来のEFM+方式では、EFM方式よりコード化レートが約6%向上するが、より高密度記録を行うためには更に高いコード化レートが望まれる。また、放送や通信などでデジタル信号を伝送する場合にも誤り率を低くした上で高いコード化レートを実現する変調方式が望まれていた。

【0006】そこで本発明は上記従来例の問題点に鑑み、8ビットデータを16ビット符号に変調するEFM+方式より更にコード化レートを向上させることができる変調方法、変調装置、復調方法、復調装置及び記録媒体を提供することを目的とする。

【0007】

【課題を解決するための手段】本発明は上記目的を達成するために、以下に示す変調方法、変調装置、復調方法、復調装置及び記録媒体を提供しようとするものである。

【0008】1. pビットの入力データ語を所定のランレンジ制限規則を満たすq（但し、 $p < q$ ）ビットの符号語に符号化すると共に、前後の符号語を直接結合しても前記所定のランレンジ制限規則を満たすように符号化する変調方法であって、入力データ語を符号語に符号化するために複数の符号化テーブルを用いると共に、前記各符号化テーブルは、それぞれの入力データ語に対応して、符号語と、この符号語に直接結合しても前記ランレンジ制限規則を満たすような次の符号語を得るために次の入力データ語を符号化するのに使用する前記符号化テーブルを示す状態情報とを格納し、さらに前記複数の符号化テーブルのうちの特定の符号化テーブルと他の特定の符号化テーブルは、所定の入力データ語に対しては、入力データ語に対応して格納されているそれぞれの符号語をNRZI変換した信号が逆極性となるように符号語が割当てられており、前記所定の入力データ語を符号化する際に前記特定の符号化テーブルが指定されている場合には、前記特定の符号化テーブルと前記他の特定の符号化テーブルとを適宜選択することにより、DSV制御を行いながら符号化するようにしたことを特徴とする変調方法。

2. $p = 8$ 、 $q = 15$ であって、前記ランレンジ制限規則は、符号語をNRZI変換した信号の最小ランレンジが3T（但し、Tは符号語のチャンネルビット周期）、最大ランレンジが11Tであることを特徴とする請求項1記載の変調方法。

【0009】3. pビットの入力データ語を所定のランレンジ制限規則を満たすq（但し、 $p < q$ ）ビットの符号語に符号化すると共に、前後の符号語を直接結合しても前記所定のランレンジ制限規則を満たすように

符号化する変調装置であって、入力データ語を符号語に符号化するための符号化テーブルを複数備え、前記各符号化テーブルは、それぞれの入力データ語に対応して、符号語と、この符号語に直接結合しても前記ランレングス制限規則を満たすような次の符号語を得るために次の入力データ語を符号化するのに使用する前記符号化テーブルを示す状態情報とを格納しており、さらに前記複数の符号化テーブルのうちの特定の符号化テーブルと他の特定の符号化テーブルは、所定の入力データ語に対しては、入力データ語に対応して格納されているそれぞれの符号語をNRZI変換した信号が逆極性となるように符号語が割当てられた構成となっていることを特徴とする変調装置。

4. pビットの入力データ語を所定のランレングス制限規則を満たすq（但し、 $p < q$ ）ビットの符号語に符号化すると共に、前後の符号語を直接結合しても前記所定のランレングス制限規則を満たすように符号化する変調装置であって、入力データ語を符号語に符号化するための符号化テーブルを複数備え、前記各符号化テーブルは、それぞれの入力データ語に対応して、符号語と、この符号語に直接結合しても前記ランレングス制限規則を満たすような次の符号語を得るために次の入力データ語を符号化するのに使用する前記符号化テーブルを示す状態情報とを格納しており、さらに前記複数の符号化テーブルのうちの特定の符号化テーブルと他の特定の符号化テーブルは、所定の入力データ語に対しては、入力データ語に対応して格納されているそれぞれの符号語をNRZI変換した信号が逆極性となるように符号語が割当てられた構成となっており、前記所定の入力データ語を符号化する際に前記特定の符号化テーブルが指定されている場合には、前記特定の符号化テーブルと前記他の特定の符号化テーブルとをDSV制御を行うように適宜選択して符号化する選択手段を備えたことを特徴とする変調装置。

5. pビットの入力データ語を所定のランレングス制限規則を満たすq（但し、 $p < q$ ）ビットの符号語に符号化すると共に、前後の符号語を直接結合しても前記所定のランレングス制限規則を満たすように符号化する変調装置であって、入力データ語を符号語に符号化するための符号化テーブルを複数備え、前記各符号化テーブルは、それぞれの入力データ語に対応して、符号語と、この符号語に直接結合しても前記ランレングス制限規則を満たすような次の符号語を得るために次の入力データ語を符号化するのに使用する前記符号化テーブルを示す状態情報とを格納しており、さらに前記複数の符号化テーブルのうちの特定の符号化テーブルと他の特定の符号化テーブルは、所定の入力データ語に対しては、入力データ語に対応して格納されているそれぞれの符号語をNRZI変換した信号が逆極性となるように符号語が割当てられた構成となっており、前の入力データ語を符号化する

際に読み出された状態情報と入力データ語とが供給され、入力データ語が前記所定の入力データ語であって、かつ状態情報が前記特定の符号化テーブルを示すか否かを判定する判定手段と、前記判定手段の判定結果に基づいて、複数又は1つの符号化テーブルから個々のバス毎に入力データ語に対応する符号語を読み出す読み出し手段と、前記バス毎に前記符号化テーブルから読み出された符号語を記憶する複数のバスメモリと、前記複数のバスメモリにそれぞれ記憶されている符号語をNRZI変換した信号のDSVの総和を記憶する複数のDSVメモリと、前記DSVの総和の絶対値が小さい方のバスメモリに記憶されている符号語を符号化データとして選択する選択手段とを、有することを特徴とする変調装置。

【0010】6. pビットの入力データ語が所定のランレングス制限規則を満たすq（但し、 $p < q$ ）ビットの符号語に符号化されていると共に、前後の符号語を直接結合しても前記所定のランレングス制限規則を満たすように符号化されているqビットの符号語をpビットの入力データ語に復調する復調方法であって、入力される符号語を復号するために入力データ語を符号化した際に用いられた複数のコードテーブルを用いると共に、前記各コードテーブルは、それぞれの入力データ語に対応して、符号語と、この符号語に直接結合しても前記ランレングス制限規則を満たすような次の符号語を得るために次の入力データ語を符号化するのに使用する前記コードテーブルを示す状態情報とを格納し、さらに前記複数のコードテーブルのうちの特定のコードテーブルと他の特定のコードテーブルは、所定の入力データ語に対しては、入力データ語に対応して格納されているそれぞれの符号語をNRZI変換した信号が逆極性となるように符号語が割当てられており、前の符号語のLSB側のゼロラン長に基づいて後の符号語の取り得る状態番号を検出すると共に、後の符号語を符号化した際に使用されたコードテーブルを示す状態番号を算出し、検出された状態番号と算出された状態番号と前の符号語とから前の符号語を復調することを特徴とする復調方法。

7. pビットの入力データ語が所定のランレングス制限規則を満たすq（但し、 $p < q$ ）ビットの符号語に符号化されていると共に、前後の符号語を直接結合しても前記所定のランレングス制限規則を満たすように符号化されているqビットの符号語をpビットの入力データ語に復調する復調装置であって、入力される符号語を復号するために入力データ語を符号化した際に用いられた複数のコードテーブルを備え、前記各コードテーブルは、それぞれの入力データ語に対応して、符号語と、この符号語に直接結合しても前記ランレングス制限規則を満たすような次の符号語を得るために次の入力データ語を符号化するのに使用する前記コードテーブルを示す状態情報とを格納し、さらに前記複数のコードテーブルのうちの特定のコードテーブルと他の特定のコードテーブル

は、所定の入力データ語に対しては、入力データ語に対応して格納されているそれぞれの符号語をNRZI変換した信号が逆極性となるように符号語が割当てられており、前の符号語のLSB側のゼロラン長に基づいて、後の符号語の取り得る状態番号を検出する検出手段と、後の符号語を符号化した際に使用されたコードテーブルを示す状態番号を算出する算出手段と、前記検出手段からの状態番号と前記算出手段からの状態番号と前の符号語とを用いて前記複数のコードテーブルを参照し、前の符号語を復調する復調手段と、を備えたことを特徴とする復調装置。

8. p ビットの入力データ語が所定のランレングス制限規則を満たす q (但し、 $p < q$) ビットの符号語に符号化されていると共に、前後の符号語を直接結合しても前記所定のランレングス制限規則を満たすように符号化されている q ビットの符号語を p ビットの入力データ語に復調する復調方法であって、後の符号語を符号化した際に使用されたコードテーブルを示す状態番号を算出して、前の符号語と後の符号語の状態番号に対応して復号データ語が蓄積されている復号テーブルを参照し、前の符号語を復調することを特徴とする復調方法。

9. p ビットの入力データ語が所定のランレングス制限規則を満たす q (但し、 $p < q$) ビットの符号語に符号化されていると共に、前後の符号語を直接結合しても前記所定のランレングス制限規則を満たすように符号化されている q ビットの符号語を p ビットの入力データ語に復調する復調装置であって、後の符号語を符号化した際に使用されたコードテーブルを示す状態番号を算出する算出手段と、前記算出手段からの状態番号と前の符号語とを用いて復号テーブルを参照し、前の符号語を復調する復調手段とを備え、前記復号テーブルは、前の符号語と後の符号語の状態番号に対応して復号データ語が蓄積されていることを特徴とする復調装置。

【0011】10. p ビットの入力データ語を所定のランレングス制限規則を満たす q (但し、 $p < q$) ビットの符号語に符号化すると共に、前後の符号語を直接結合しても前記所定のランレングス制限規則を満たすように符号化された符号語がNRZI変換されて記録されている記録媒体であって、入力データ語を符号語に符号化するために複数の符号化テーブルを用いられると共に、前記各符号化テーブルは、それぞれの入力データ語に対応して、符号語と、この符号語に直接結合しても前記ランレングス制限規則を満たすような次の符号語を得るために次の入力データ語を符号化するのに使用する前記符号化テーブルを示す状態情報とを格納し、さらに前記複数の符号化テーブルのうちの特定の符号化テーブルと他の特定の符号化テーブルは、所定の入力データ語に対しては、入力データ語に対応して格納されているそれぞれの符号語をNRZI変換した信号が逆極性となるように符号語が割当てられており、前記所定の入力データ語を

符号化する際に前記特定の符号化テーブルが指定されている場合には、前記特定の符号化テーブルと前記他の特定の符号化テーブルとを適宜選択することにより、DSV制御を行いながら符号化された符号語がNRZI変換されて記録されていることを特徴とする記録媒体。

11. $p = 8$ 、 $q = 15$ であって、前記ランレングス制限規則は、符号語をNRZI変換した信号の最小ランレングスが3T (但し、Tは符号語のチャンネルビット周期)、最大ランレングスが1.1Tであることを特徴とする請求項10記載の記録媒体。

【0012】

【発明の実施の形態】以下、図面を参照して本発明の実施の形態を説明する。図1は本発明の変調装置を使用したディスク記録装置の例を示す概略構成図、図2は本発明に係る変調装置の第1の実施形態を示すブロック図、図3はRL(2, 10)の符号化器の状態遷移を示す説明図、図4は15状態の符号化テーブルの一部を示す説明図、図5は図4に示した符号化テーブルを新たな符号化テーブルに変換する変換方法を説明するための説明図、図6～図12は本発明に係る新たな符号化テーブルの第1の実施形態を示す説明図、図13は符号化処理を概略的に説明するためのブロック図、図14は図2の変調装置の符号化処理を示すフローチャート、図15は本発明に係る復調装置を使用したディスク再生装置の例を示す概略構成図、図16は本発明に係る復調装置の第1の実施形態を示すブロック図、図17は状態演算器の処理を示す説明図、図18～図19は状態の取りうる値を説明するための図、図20は本発明に係る復調装置の復調処理を示すフローチャート、図21は図16は図21は本発明に係る復調装置の第2の実施形態を示すブロック図、図22は図21の復号テーブルのケーステーブルの一部を示す説明図である。

【0013】図1は本発明の変調装置を使用したディスク記録装置の例を示す概略構成図である。同図において、ディスク記録装置1は、映像や音声などのデジタル信号を記録媒体2に記録する装置である。そして、映像や音声などのデジタル信号と一緒に記録する制御信号等と共にフォーマット部12に入力され、ここでECCや同期信号などが付加されて記録媒体2の記録フォーマットに合わせた物理フォーマットに変換され、ソースコードとして8-15変調器(変調装置)13に出力される。8-15変調器13では、入力されるソースコードを8ビットごとにコードテーブル120に対応させ、この8ビットのソースコードを15ビットに変換して順次出力する。そして、この8-15変調器13から出力された信号はNRZI変換回路14に入力され、ここでNRZI変換されて記録媒体駆動回路15により記録媒体2に記録される。ここで、8-15変調器13は、本願発明の変調装置の一実施の形態であり、例えば図2に示すような構成となっている。そして、この8-15変調

器13に含まれるコードテーブル120の構成について、以下に説明する。図3はチャンネルビットをTとして、最小ランレングスが3T、最大ランレングスが11Tに制限されるRLL(2, 10)の符号化器の状態遷移を示している。図の円内に示されている「0」から「10」の数字は、この符号化器の内部状態に相当し、円から出力されている矢印に示されている「0」、「1」の数字は、この符号化器の内部状態が遷移した際に出力される符号語に相当する。したがって、図1によれば、「1」と「1」の間の「0」の数が2個以上、10個以下になるRLL(2, 10)の規則を守る符号語が生成される。次にこの符号化器からRLL(2, 10)の性質を持ち、8ビットのデータ語を15ビットの

符号語に変換が可能であることを説明する。図3に示するような状態遷移を行う符号化器から、本符号化器の隣接行列Aが求まる。ここで隣接行列とは、状態遷移図中の各状態の依存関係を示す行列であり、第1行～第n行に遷移前の状態0～状態(n-1)を割当て、第1列～第n列に遷移後の状態0～状態(n-1)を割当てて、例えば、状態0から状態1にパスがある場合を1、無い場合を0とした行列である。そして、図1のRLL(2, 10)符号化器は11状態あるので、隣接行列Aは11次の行列(式1)で表現することができる。

【0014】

【数1】

$$A = \begin{pmatrix} 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix} \quad \dots (1)$$

【0015】ここで符号化限界レートは符号化器の状態遷移図から求まる隣接行列の固有値の中の最大値の2の対数で求まることが公知であり、式(1)の最大固有値λ(max)が1.4558であることから

$$\log_2(1.4558) = 0.54$$

と求まる。すなわち、RLL(2, 10)の符号化限界のコードレートは0.54であり、このレート以下の符号化レートでの符号化器の構成は可能である。本式より、p-qコードでp=8とした場合、qは15が限界となることがわかる(この場合、符号化レートは8/15=0.533...となる)。したがって、RLL(2, 10)で8-q符号を作る場合、q=15より小

の符号化器は成り立たないが、15であれば可能であることがわかる。

【0016】次に実際の符号化器の構成について述べる。最も簡単な構成は図1の符号化器をそのまま用いればよいが、その場合には8ビットの入力語に対応する出力として各状態から256(2の8乗)以上のパスが出力され、その出力先でも256通り以上の出力パスが必要となる。これを確認するにはAの15乗を取り、各状態から何本のパスが生成されるかを見ればよい。

【0017】

【数2】

```
00: 057 040 027 018 013 009 006 004 003 002 001 SUM = 180 (256)
01: 083 057 040 027 018 013 009 006 004 003 002 SUM = 262
02: 122 083 057 040 027 018 013 009 006 004 003 SUM = 382
03: 120 082 056 039 027 018 012 009 006 004 003 SUM = 376
04: 117 080 055 038 026 018 012 008 006 004 003 SUM = 367
05: 113 077 053 037 025 017 012 008 005 004 003 SUM = 354 ... (2)
06: 107 073 050 035 024 016 011 008 005 003 003 SUM = 335
07: 098 067 046 032 022 015 010 007 005 003 002 SUM = 307
08: 085 058 040 028 019 013 009 006 004 003 002 SUM = 267
09: 067 045 031 022 015 010 007 005 003 002 002 SUM = 209 (256)
10: 040 027 018 013 009 006 004 003 002 001 001 SUM = 124 (256)
```

【0018】式(2)は式(1)の15乗の値と各状態

(00:~10:と記す)から何本のパスが生成される

か (SUM=xと記す) を計算した結果を示すものである。

【0019】この式(2)によると、256通りの8ビットデータ (=0~255) に対して、上記の状態「0」、「9」及び「10」に対応する符号語の数(パスの数)は、足りないことになる。しかしながら、公知例「ファイナリティ ステート モジュレーション コード フォー データ ストレージ」"Finite-State Modulation Codes for Data Storage", B. H. Marcus他、IEEE Journal on selected areas in Communication, vol. 10, No.1, 1992年1月に説明されているように、状態を分割して選別することにより、15の状態からなる符号化テーブルを作成することができる。ここで、この実施形態では第1の例として、図3における状態「0」、「1」、「6」、「8」、「9」、「10」を選択し

$$T = \begin{pmatrix} 57 & 40 & 73 & 7 & 2 & 1 \\ 83 & 57 & 107 & 10 & 3 & 2 \\ 107 & 73 & 136 & 13 & 3 & 3 \\ 85 & 58 & 109 & 10 & 3 & 2 \\ 67 & 45 & 85 & 8 & 2 & 2 \\ 40 & 27 & 50 & 5 & 1 & 1 \end{pmatrix} \cdots (3)$$

【0023】この式(3)は、状態「0」から状態「0」へ遷移するパスが57本、状態「0」から状態「1」へ遷移するパスが40本、状態「0」から状態「6」へ遷移するパスが73本、…、状態「1」から状態「0」へ遷移するパスが83本、…、状態「10」から状態「10」へ遷移するパスが1本であることを示している。そして、このように、各状態を2, 3, 4, 3, 2, 1通りに分割する場合、ベクトルVを列ベクトルとして、

状態「0」	:	5 5 2 通り (行列Tの第1行の和: 57+40+73+7+2+1)
状態「1」	:	8 0 3 通り (行列Tの第2行の和: 83+57+107+10+3+2)
状態「6」	:	1 0 2 5 通り (行列Tの第3行の和: 107+73+136+13+3+3)
状態「8」	:	8 1 8 通り (行列Tの第4行の和: 85+58+109+10+3+2)
状態「9」	:	6 3 9 通り (行列Tの第5行の和: 67+45+85+8+2+2)
状態「10」	:	3 7 9 通り (行列Tの第6行の和: 40+27+50+5+1+1)

【0025】そして、これは式(5)を満たしているので、符号化テーブルの構成が可能である。ここで以上で述べた分割によって、8ビットの入力データ語を15ビットの出力符号語に変換するための符号化テーブルを作成する。

【0026】そして、例えば状態「0」を2つに分割した状態をそれぞれ「01」、「02」としたとき、状態「01」、「02」が同じ出力符号語を持つと復号することができないので、重複する出力符号語を持たないように割り当てる。同様に、他の状態から生成される出力

て、

【0020】

【表1】状態「0」 : 2通り

状態「1」 : 3通り

状態「6」 : 4通り

状態「8」 : 3通り

状態「9」 : 2通り

状態「10」 : 1通り

に分割する。

【0021】状態「0」、「1」、「6」、「8」、「9」、「10」を選択したことにより、式(2)は、次式(3)に再構成される。

【0022】

【数3】

$V = (2, 3, 4, 3, 2, 1) \cdots (4)$

として、

$TV \geq 256V \cdots (5)$

であれば、符号化テーブルは生成可能であり、状態「0」、「1」、「6」、「8」、「9」、「10」をそれぞれ2, 3, 4, 3, 2, 1に分割したことにより、表2に示すようなパスが得られる。

【0024】

【表2】

符号語に対して分割を行うことにより、新たな15の状態から成る符号化テーブルを構成することができる。図4は以下に示す15の状態から成る符号化テーブルの内、元の状態が状態「0」に遷移する符号語(式(3)の第1列)の一部を10進数で示す。なお、図4において、横軸の「01」~「10」は分割後の元の状態を示し、縦軸の「001」~「010」…は、入力語を示す。そして、交叉したところの値が状態「0」に遷移したときの符号語を示している。また、図示していない残りの状態「0」に遷移する符号語、及び状態「1」、状

態「6」、状態「8」、状態「9」、状態「10」に移る符号語の表も図4と同様な構成となっている。

【0027】

【表3】状態「0」→「01」、「02」

状態「1」→「11」、「12」、「13」

状態「6」→「61」、「62」、「63」、「64」

状態「8」→「81」、「82」、「83」

状態「9」→「91」、「92」

状態「10」→「10」

【0028】しかしながら、このままではテーブル数が多く、変調・復調処理が複雑になるので、本発明では、各状態に対して一部重複する符号語を含むが、8ビットの入力データ語に対して256通りの出力符号語を有し、かつ復号可能なように状態数を削減する。具体的には、図4において、横軸として示されている分割後の元の状態「02」と状態「12」、状態「62」は同じ入力語「001」…に対しては同じ符号語「01025」…を出力するように構成されている。同様に、分割後の元の状態「13」、状態「61」、状態「81」は同じ符号語を出力し、分割後の元の状態「64」、状態「83」、状態「92」、状態「10」は同じ符号語を出力している。したがって、図5に示すように、同じ符号語を出力する分割後の元の状態をまとめて新たな状態とすると、状態「0」～状態「6」の7つの状態に削減することができる。

【0029】なお、7つの符号化テーブル数とするのは、復号時に行う先読みのバイト数をできるだけ最小に制限するためであり、状態数(符号化テーブル数)は更に削減することも可能である。

【0030】図6～図12は図5に示したようにして状態数削減を行った後の各状態「0」～「6」のテーブルをそれぞれ示している。なお、入力語は10進(「000」～「255」)で示され、変換後の出力符号語は2進(15ビット)で示されている。また、出力符号語の右側の数字は、符号語の間を直接に結合しても、ランレングス制限規則を満足するための次の状態「0」～「6」を示す。ここで、例えば図6に示す状態「0」のテーブルを参照すると、入力語「000」、「001」…に対して以下のように同一の出力符号語が対応している。

【0031】

【表4】(i) 入力語「000」に対して

000000000100000 次の状態「4」

入力語「001」に対して

000000000100000 次の状態「5」

(ii) 入力語「002」に対して

000000000100001 次の状態「0」

入力語「003」に対して

000000000100001 次の状態「1」

...

【0032】このように出力符号語は各状態で重複しているが、どの入力データ語に対応しているかは、次に遷移する状態により決定されている。また、ある符号語に対して、次に遷移する状態に含まれる符号語は必ず独立している(次にとる状態間では重複する符号語は存在しない)ように割当てているので、復号時に元の入力データ語を確定することができる。

【0033】また、状態「0」と状態「3」の各テーブルに着目すると、入力データ語「000」～「038」に対応する各出力符号語をNRZI変調した信号は、極性が逆(符号語に含まれる「1」の数の偶奇性が異なる)となるように配置されている。符号化規則については、各テーブルとも次に遷移する状態として状態「0」が選択される場合は、前の出力符号語のLSB側のゼロラン長が「0」の場合(すなわち出力符号語が「1」で終わる場合)となっている。また、状態「3」のテーブルにおいては、入力データ語「000」～「038」に対応する各出力符号語は、MSB側のゼロラン長が「2」となるように(すなわち出力符号語が「001」で始まるように)配置されている。したがって、状態「3」のテーブルの入力データ語「000」～「038」に対応する各出力符号語をそれぞれ状態「0」のテーブルにおける入力データ語「000」～「038」に対応する各出力符号語と交換しても、NRZI変調後のラン長が3T～11Tに限定される符号化規則を維持することができる。また、この出力符号語の交換を行っても、1つ前の出力符号語の復号には何ら問題は生じない。

【0034】同様に、状態「2」と状態「4」の各テーブルに着目すると、入力データ語「000」～「011」及び「026」～「047」に対応する各出力符号語をNRZI変調した信号は、同じく極性が逆になるように配置されている。符号化規則については、各テーブルとも次に遷移する状態として状態「2」が選択される場合は、前の出力符号語のLSB側のゼロラン長が「1」の場合(すなわち出力符号語が「10」で終わる場合)となっている。また、状態「4」のテーブルにおいては、入力データ語「000」～「011」及び「026」～「047」に対応する各出力符号語は、MSB側のゼロラン長が「1」となるように(すなわち出力符号語が「01」で始まるように)配置されている。

【0035】したがって、状態「4」のテーブルの入力データ語「000」～「011」及び「026」～「047」に対応する各出力符号語をそれぞれ状態「2」のテーブルにおける入力データ語「000」～「011」及び「026」～「047」に対応する各出力符号語と交換しても、NRZI変調後のラン長が3T～11Tに限定される符号化規則を維持することができ、かつ1つ前の出力符号語を復号することができる。なお、状態「2」における入力データ語「012」～「025」

に対応する出力符号語が状態“4”では割り当てられていないのは、状態“0”、“2”の符号語の並びが崩れて復号できなくなることを防止するためである。そして、各状態テーブルは全ての対応する出力符号語の極性が逆になっているわけではないが、なるべく極性が逆になるように配置されている。

【0036】また、図6～図12に示す符号化テーブルの例では、15ビット出力符号語のLSB側のゼロラン長に関連して以下に示す4種類のケース「0」～「3」に大別することができる。

【0037】

【表5】ケース「0」：LSB側のゼロラン長は0

ケース「1」：LSB側のゼロラン長は1

ケース「2」：LSB側のゼロラン長は2から6

ケース「3」：LSB側のゼロラン長は7から10

【0038】そして、各ケース「0」～ケース「3」において、各出力符号語に対して次に遷移する状態は次のような関係にある。

【0039】

【表6】ケース「0」→状態“0”、“1”

ケース「1」→状態“1”、“2”、“3”

ケース「2」→状態“1”、“3”、“4”、“5”

ケース「3」→状態“3”、“4”、“5”、“6”

【0040】但し、これに属さない、例えば1種類しかない出力符号語については、次に状態が存在する種類の内のどれかに割り当てる。

【0041】このような符号化テーブルに対して、例えば初期状態“0”において入力データ語が「000」→「001」→「002」の順で入力される場合には、バス「1」、「2」について次のような処理を行う。

【0042】

【表7】(1)バス「1」

(1-1) 入力データ語が「000」のとき出力符号語として状態“0”のテーブルの0000000001000000が選択され、次の状態は“4”に移行する。

(1-2) 入力データ語が「001」のとき出力符号語として状態“4”のテーブルの0100010000000000が選択され、次の状態は“5”に移行する。

(1-3) 入力データ語が「002」のとき出力符号語として状態“5”のテーブルの100001000000100000が選択され、次の状態は“1”に移行する。

【0043】

【表8】(2)バス「2」

(2-1) 入力データ語が「000」のとき出力符号語として状態“3”のテーブルの0010010000000000が選択され、次の状態は“4”に移行する。

【0044】その後、入力データ語が「001」、「002」の順で入力されると、バス「0」の場合と同じ出力符号語が選択され、同じ状態に移行することになる。

図13はその処理を概略的に示すものであり、符号化テ

ーブル120内には図6～図12に示したような符号化テーブルが記憶されている。そして、符号化テーブル120から出力される出力符号語は、今回の入力データ語と「次の状態」に基づいて選択されることになる。

【0045】図2は符号化装置(変調装置)13の構成を概略的に示す図である。なお、図2は符号を一時記憶するためのバスメモリが2つの場合を示しているが、本発明はより多くのバスメモリを有する場合にも適用することができる。まず、同期信号などの入力データ語SC1に対して初期テーブル(符号化テーブル120の選択肢の初期値)を選択しておく。次いで8ビットの入力データ語SCtが入力されると、符号語選択肢有無検出回路100は今回の入力データ語SCtと、符号化テーブルアドレス演算部110から供給される先行出力符号語(ここでは選択された初期値)によって決定された状態とに基づいて今回の出力符号語が一意に決まるか、または選択肢があるかを検出し、検出結果を符号化テーブルアドレス演算部110と絶対値比較部140に出力する。

【0046】符号化テーブルアドレス演算部110はこの検出結果に基づいて符号化テーブル120のアドレスを算出する。ここで、符号語選択肢有無検出回路100の検出結果が「選択肢あり」の場合には、符号化テーブルアドレス演算部110により算出されるアドレスは2つとなるので、この場合には、符号化テーブル120は時分割処理などにより2種類の符号語を出力する。そして、符号化テーブル120から出力される2種類の符号語は、一方がバス「1」の出力符号語としてバスメモリ131に入力され、他方がバス「2」の出力符号語としてバスメモリ133に入力される。また、符号語選択肢有無検出回路100の検出結果が「選択肢なし(一意に決まる)」の場合には、符号化テーブルアドレス演算部110により算出されるアドレスは1つであるので、このアドレスに対応する出力符号語が符号化テーブル120から読み出されて、バスメモリ131、133に同じ出力符号語が入力される。

【0047】ここで、バスメモリ131には、過去に出力された以降の出力符号語と直前に入力されたバス「1」の出力符号語が蓄積されており、バスメモリ133には、過去に選択された全ての出力符号語と直前に入力されたバス「2」の出力符号語が蓄積されている。また、DSV演算メモリ130には、過去に選択された全ての出力符号語と直前に入力されたバス「1」の出力符号語から得られるDSV値が記憶されており、DSV演算メモリ132には、過去に出力された以降の出力符号語と直前に入力されたバス「2」の出力符号語から得られるDSV値が記憶されている。そして、DSV演算メモリ130、132に記憶されている各DSVは、絶対値比較部140に供給されてその絶対値|DSV|を比較し、符号語選択肢有無検出回路100の検出結果が

「選択肢あり」の場合には、その比較結果をメモリ制御／符号出力部150に出力する。そして、絶対値 $|DSV|$ の小さい方のパスを選択して、絶対値 $|DSV|$ の大きい方のパスメモリ131又は133の内容とDSV演算メモリ130又は132の内容をそれぞれ絶対値 $|DSV|$ の小さい方のパスメモリ133又は131の内容とDSV演算メモリ132又は130の内容で書き換える。その後、パスメモリ131では今回入力されてきたパス「1」の出力符号語も合わせて保持すると共に、今回入力されてきたパス「1」の出力符号語はDSV演算メモリ130にも出力して新たなDSVを演算し記憶する。同様に、パスメモリ133では、今回入力されてきたパス「2」の出力符号語も合わせて保持すると共に、今回入力されてきたパス「2」の出力符号語はDSV演算メモリ132にも出力して新たなDSVを演算し記憶する。なお、符号語選択肢有無検出回路100の検出結果が「選択肢なし」の場合には、パス「1」及びパス「2」の出力符号語は同じとなる。以上の動作を入力データ語が無くなるまで繰り返し、最後にパスメモリ131又は133に蓄積されている全ての出力符号語を出力することにより、NRZI変換後に3T～11Tを満足するDSV制御された出力符号語を出力することができる。

【0048】次に、図14に示すDSV制御のフローチャート図を参考にしながらその動作の具体例について詳しく説明する。まず、同期信号などの入力データ語SC1に対して初期テーブル（符号化テーブル120の選択肢の初期値）を選択する（ステップ401）。次いで8ビットの入力データ語SCtが入力されると（ステップ403）、符号語選択肢有無検出回路100は今回の入力データ語SCtと、符号化テーブルアドレス演算部110から供給される先行出力符号語（最初の場合は選択された初期値）によって決定された状態とに基づいて今回の出力符号語が一意に決まるか、または選択肢があるかを検出し（ステップ405）、検出結果を符号化テーブルアドレス演算部110と絶対値比較部140に出力する。ここで、図6～図12に示す符号化テーブルにおいて状態“0”と状態“3”に着目すると、上述の符号化テーブルのところで説明したように、状態“3”の出力符号語の内、入力データ語「000」～「038」に対応する出力符号語は、状態“0”の出力符号語と交換しても符号化規則を維持することができ、また、復号可能である。また、状態“2”と状態“4”に着目すると、状態“4”の出力符号語の内、入力データ語「000」～「011」及び「026」～「047」に対応する出力符号語は、状態“2”の出力符号語と交換しても符号化規則を維持することができ、また、復号可能である。さらに、図6～図12に示す符号化テーブルでは、状態“0”と状態“2”の出力符号語は、それぞれ状態“3”と状態“4”の前記した入力データ語に対応する

出力符号語においてはNRZI変換後の極性が逆になるように構成されている。このため、状態“0”における入力データ語「000」～「038」と、状態“2”における入力データ語「000」～「011」及び「026」～「047」が発生した場合には、複数の出力符号語が採りうることになり、パス「1」、パス「2」としてDSVの値を利用して最適な出力符号語を選択することによりDSV制御を行うことが可能となる。

【0049】そこで、符号選択肢有無検出回路100において、符号化テーブルアドレス演算部110から供給される状態が状態“0”であって、入力データ語SCtが「038」以下の場合には（ステップ407→Yes）、「選択肢あり」の検出結果を出力し、符号化テーブルアドレス演算部110は、符号化テーブル120から状態“0”のテーブルの入力データ語SCtに対応する出力符号語OC1tを読み出すと共に状態“3”のテーブルの入力データ語SCtに対応する出力符号語OC2tを読み出す（ステップ409）。そして、DSV演算メモリ130、132に記憶されているそれぞれのDSV（DSVの総和）の絶対値 $|DSV|$ を絶対値比較部140にて比較する（ステップ419）。ここで、DSV演算メモリ130からのDSV1t-1の絶対値 $|DSV1t-1|$ の方が小さい場合には（ステップ419→Yes）、パスメモリ131に蓄積されている過去の出力符号語をパスメモリ133に出力して書き換えると共に、DSV演算メモリ130に記憶されているDSV1t-1でDSV演算メモリ132を書き換える（DSV演算メモリ132の内容をDSV1t-1にする）（ステップ420）。また、DSV演算メモリ132からのDSV2t-1の絶対値 $|DSV2t-1|$ の方が小さい又は同じである場合には（ステップ419→No）、パスメモリ133に蓄積されている過去の出力符号語を出力すると共に、DSV演算メモリ132に記憶されているDSV2t-1でDSV演算メモリ130を書き換える（DSV演算メモリ130の内容をDSV2t-1にする）（ステップ421）。ステップ420及び421の後、パス「1」の出力符号語OC1tをパスメモリ131に追加記憶させると共に、パス「2」の出力符号語OC2tをパスメモリ133に追加記憶させる（ステップ422）。そして、パス「1」の出力符号語OC1tを含めたDSVをDSV演算メモリ130で演算して記憶すると共に、パス「2」の出力符号語OC2tを含めたDSVをDSV演算メモリ132で演算して記憶する（ステップ423）。そして、次の入力データ語がある場合には（ステップ424→No）ステップ403に戻り、次の入力データ語がなくなった場合には（ステップ424→Yes）パスメモリ131（又はパスメモリ133）に記憶されている出力符号語のデータ列をメモリ制御／符号出力部150から出力する（ステップ425）。ステップ407において、符号選択肢有無検出回路100が、符号化テ-

ブルアドレス演算部110から供給される状態が状態“0”かつ、入力データ語SCtが「038」以下でない場合には(ステップ407→No)、さらに、状態“2”であって、入力データ語SCtが「011」以下又は「026」～「047」の範囲にある可否かを判断し(ステップ411)、適合する場合には(ステップ411→Yes)、「選択肢あり」の検出結果を出力し、符号化テーブルアドレス演算部110は、符号化テーブル120から状態“2”のテーブルの入力データ語SCtに対応する出力符号語OC1tを読み出すと共に状態“4”のテーブルの入力データ語SCtに対応する出力符号語OC2tを読み出す(ステップ413)。そして、上記した場合と同様、ステップ419～425の処理を行う。

【0050】さらに、ステップ411において適合しない場合には(ステップ411→No)、前に出力された出力符号語が前述した「0」～「3」のどのケースに属するかを判断する(ステップ415)。具体的には、ある符号語に対して次に遷移する状態は、前述したように4種類のケース「0」～「3」に大別することができる。そして、ある入力データ語に対応する出力符号語については、前の出力符号語がケース「2」(LSB側のゼロラン長が2から6)に含まれ、次の出力符号語が状態“1”、“3”、“4”、“5”のうち状態“3”の符号化テーブルから選択される出力符号語である場合には、符号化規則を崩さない範囲で状態“0”の符号化テーブルにおける出力符号語と入れ替えることができる。したがって、ステップ415においては、前の出力符号語がケース「2」(LSB側のゼロラン長が2から6)に含まれ、次の出力符号語が状態“3”の符号化テーブルから選択される出力符号語であって、状態“0”の符号化テーブルにおける出力符号語と入れ替えても符号化規則を崩さない範囲にあるかどうかを判断し、適合する場合には(ステップ415→Yes)、「選択肢あり」の検出結果を出力し、符号化テーブルアドレス演算部110は、符号化テーブル120から状態“3”のテーブルの入力データ語SCtに対応する出力符号語OC1tを読み出すと共に状態“0”のテーブルの入力データ語SCtに対応する出力符号語OC2tを読み出す(ステップ417)。そして、上記した場合と同様、ステップ419～425の処理を行う。

【0051】また、適合しない場合には(ステップ415→No)、「選択肢なし」の検出結果を出力し、符号化テーブルアドレス演算部110は、符号化テーブル120から入力データ語SCtに対応する出力符号語OC1tを読み出して、バスメモリ131、133に出力して保持する(ステップ418)。この場合、バス「1」、「2」の出力符号語OC1t、OC2tの値は同じとなる。そして、上記した場合と同様、ステップ423～425の処理を行う。なお、ステップ415→N

oすなわち、「選択肢なし」の場合には、DSVの絶対値の比較やバスの選択などは行わず、「選択肢あり」となるまでバスメモリ131、133への蓄積及びDSV演算メモリ130、132でのDSV算出更新のみを行っている。

【0052】そして、このようにして符号化された15ビットの出力符号語がNRZI変換されると、このNRZI変換信号は最小ランレングスが3T(T=チャンネルビットの周期)、最大ランレングスが11Tの規則を満たし、光ディスクなどの媒体に高密度で記録することができる。また、空中や伝送ケーブルなどで伝送する場合にも少ないデータ量で誤りなく伝送することができる。

【0053】次に図15を参照しながら復調装置(復号装置)の一例について説明する。ここで、上記のように8-15変調されて前後の出力符号語(以降単に符号語という)が結合された符号系列を復号する場合、

- ・先行する符号語と、
- ・先行する符号語がどのケースに属しているかと、
- ・後続の符号語がどの状態の符号化テーブルから生成されているか

を特定することにより復号することができる。

【0054】図15は、例えば図1に示したようなディスク記録装置によって情報信号が記録された光ディスク2から記録された情報を再生するディスク再生装置3の例を示す概略構成図である。同図において、ディスク再生装置3は、記録媒体2に記録されている情報信号を記録媒体駆動装置31により再生してNRZI復号器32に出力し、ビットクロックと15ビットの符号語が結合されている符号系列が復号される。そして、15-8復調器(復調装置)33で、ビットクロックを使用して15ビットの符号語から8ビットのソースコードが復調され、デフォーマット部34に供給されてECCや同期信号などが取り除かれ、コンテンツ情報を示すデジタル信号が出力される。次に、このようなディスク再生装置3で使用される復調装置33のいくつかの実施の形態について以下に説明する。図16は、復調装置33の一実施の形態を示す構成図である。同図においてNRZI復調器32から供給される再生符号系列は、同期検出回路300とシリアル/パラレル変換器301とに供給され、ビットクロックは、同期検出回路300とシリアル/パラレル変換器301及びゼロラン長検出回路303に供給される。そして、同期検出回路300では再生符号系列を15ビット符号語単位で再構成するためのワードクロックが生成され、このワードクロックはシリアル/パラレル変換器301と、状態演算器302とゼロラン長検出回路303とワードレジスタ304とに供給される。

【0055】シリアル/パラレル変換器301では再生符号系列がワードクロックに基づいて15ビット符号語単位で再構成され、この15ビット符号語は、ワードレ

レジスタ304と状態演算器302に供給される。ワードレジスタ304では15ビット符号語がワードクロックに基づいて1ワード分(15ビット分)だけ遅延され、前の符号語(参照アドレス)としてゼロラン長検出回路303及びコード復号部305に供給される。

【0056】ゼロラン長検出回路303はビットクロックとワードクロックを用いて、ワードレジスタ304からの15ビット符号語のLSB側のゼロラン長を検出してケース情報をコード復号部305に出力する。ここで、前述したように図6～図12に示した符号化テーブルでは、15ビット符号語のLSB側のゼロラン長と、ケース「0」～「3」は次のような関係にある。

【0057】

【表9】ケース「0」：LSB側のゼロラン長は0

ケース「1」：LSB側のゼロラン長は1

ケース「2」：LSB側のゼロラン長は2から6

ケース「3」：LSB側のゼロラン長は7から10

【0058】したがって、15ビット符号語Ck-1のLSB側のゼロラン長を検出して、この検出結果と後続の符号語Ckがどの状態“0”～“6”であるかを判定することにより、復号データを一意に決定することができる。

【0059】そこで、状態演算器302では後続の符号語Ckがどの状態の符号化テーブルにより符号化されたかを示す情報として、図17にC言語で示すような演算式が正となる式と対応する状態番号を示す状態情報(図中、各演算式に対して左端の番号0～6がそれぞれ対応し、さらに符号化テーブルに付された番号“0”～

“6”に対応する)をコード復号部305の状態レジスタ310に出力する。なお、図17中、Ckは15ビットの符号語を10進で示したものである。ここで、図17で示される演算式により得られる状態番号について、図18及び図19を参照しながら説明する。各図において、10進及び2進で示されている各15ビットの符号語Ckに対して、図17で示される演算式により得られる“0”～“6”の状態番号に対応する7つのカラムが示されている。各カラムにおいて、太線及び黒丸は符号語Ckに対して取り得る状態番号の範囲を示している。太線上の白丸は太線の範囲内において除外されている符号語Ckの値(該当状態番号には対応しない符号語Ckの値)を示している。同図では、所定の符号語Ckに着目すると、複数の太線が対応している場合があり、所定の符号語Ckに対して複数の状態番号が取り得る、すなわち、状態演算器302からは複数の状態番号が出力されることを示している。したがって、コード復調部305は、状態演算器302から供給される状態情報とゼロラン長検出回路303から供給される1ワード前の符号語Ck-1のケース情報とを参照して取り得る状態番号を確定する。そして、このコード復号部305で使用するコードテーブル120は、図1及び図2に示した変調装

置13の符号化テーブル(図6～図12参照)120と同じコードテーブルであり、状態レジスタ310は1ワード前の符号語Ck-1の状態番号を記憶しているので、確定した符号語Ckの状態番号とワードレジスタ304から供給される1ワード前の符号語(参照アドレス)に基づいてコードテーブル120の符号語Ck-1の状態番号が示すサブテーブルを参照し、該当する8ビットの復号データ語(ソースコード)を出力する。例えば、コード復調部305がゼロラン長検出回路303からケース「2」を示すケース情報を受け取った場合には、取り得る状態は状態“1”、“3”、“4”、“5”であり、図19に斜線で示す範囲となる。この中で太線及び黒丸で示される範囲が得られる15ビットの符号語Ckの範囲である。そして、符号語Ckが、 $Ck \leq 585$ 、または $1024 \leq Ck < 9216$ 、または $9216 < Ck$ である場合には、取り得る状態番号は1つに限定されることになる。また、次の符号語Ckの状態番号が1つの状態に限定されない場合でも、コードテーブル120(図6～図12参照)の符号語Ck-1の状態番号に対応するサブテーブル内では、15ビットの符号語Ck-1と複数ある次の符号語Ckの状態番号との各組み合わせのうち、格納されている組合せは1つだけであるので、このサブテーブルに格納されている組み合わせに該当する状態番号を次の符号語Ckの状態番号として確定させることができる。このようにして符号語Ckの状態番号と前の符号語Ck-1のケース情報とから前の符号語Ck-1を復号していくことができる。このコード復調部305の動作について、図20に示すフローチャートと共に説明する。まず、状態レジスタ310を初期値0にする(ステップ401)。次に、ワードレジスタから符号語Ck-1を得る(ステップ403)。そして、ゼロラン長検出回路303から符号語Ck-1のケース情報を得る(ステップ405)。このケース情報を使用して取り得る状態番号を検出する(ステップ407)。状態演算器302から次の符号語Ckの状態情報を得る(ステップ409)。検出した取り得る状態番号と次の符号語Ckの状態情報が示す状態番号とに共通な状態番号を得ることにより、次の符号語Ckの状態番号を確定する(ステップ411)。状態レジスタ310に記憶されている符号語Ck-1の状態番号(初期値の場合は“0”)の示すコードテーブル120のサブテーブル(符号語Ck-1の状態番号が示す番号のサブテーブル)内を検索して、確定した次の符号語Ckの状態番号と符号語Ck-1とが格納されているソースコードを検出する(ステップ413)。検出した8ビットのソースコードを符号語Ck-1の復号コード語として出力する(ステップ415)。次の符号語Ckを復号するために、次の符号語Ckの状態番号を状態レジスタ310に記憶する(ステップ417)。以下、復号する符号語が無くなるまでステップ403以降を繰り返す。

【0060】次に、図21を参照しながら本発明の復調装置の第2の実施の形態について説明する。図21に示す復調装置33aの構成は図16に示す復調装置33の構成と比較すると、ケース情報を出力するゼロラン長検出回路303が省略され、コード復調部305aには、コードテーブル120の代わりに図22に示すような15ビットの符号語に対応する復号データを記憶しているコードテーブル306が備えられている。なお、他の構成（同期検出回路300、シリアル／パラレル変換器301、状態演算器302、ワードレジスタ304）は同じ構成であるので、その説明は省略する。

【0061】そして、図21に示す復調装置33aのコード復調部305aには、ワードレジスタ304から前の符号語Ck-1が供給されると共に状態演算器302からは現時点の符号語Ckの状態情報が供給される。ここで、図22は、コード復調部305aに備えられているコードテーブル306の一部を示すものであり、入力される15ビットの符号語（10進及び2進で示す）Ck-1を参照アドレスとして、状態演算器302から供給される状態情報（図17に示す演算結果による後続符号語の状態）“0”～“6”に対応する10進の数字（8ビットの復号データ）を出力するものである。なお、後述する説明をわかりやすくするために、符号語に対応する

000000000100000 (10進で「32」)
010001000000000 (10進で「8704」)
100001000001000 (10進で「16904」)

【0066】LSB側のゼロラン長またはコードテーブル306から、最初の符号語「000000000100000」に対応するケースはケース「2」であることが分かる。そして、次の符号語が状態“1”、“3”、“4”、“5”の内のどの状態に遷移するかを見ればこの符号語を復号することができる。

【0067】この場合には次の符号語「010001000000000」が状態“4”になることから、状態“4”に対応する復号データ語を出力すれば良い。図22に示すコードテーブル306を参照すると、「0」が8ビットの復号データ語として出力される（符号化された元のデータは「0」であったことが判る）。同様に2番目の「010001000000000」はケース「2」の符号であり、次の符号語の状態は“5”であることから、コードテーブル306を参照すると、8ビットの復号データ語は「1」であることが判る。したがって、この復号方式によれば、現在の符号語と次の符号語に基づいて現在の符号語（又は前の符号語と現在の符号語に基づいて前の符号語）を元のデータに復号することができる。なお、この実施の形態において、ケースを使用して説明したのは、ケースによっては、次の符号語の状態のうち取り得ない状態が存在し、その取り得ない状態に対応する復号コードはコードテーブル306に記憶する必要が無いことを説明するためである。そして、入

ケース情報も記載しているが、実際のコード復調部305aに備える場合にはケース情報はなくても良い。そして、このようなコードテーブル306を使用する場合には、前の符号語と現時点の符号語の状態が判れば、前の符号語は復号可能である。

【0062】図21に示した復調装置33aを用いた場合の具体的な復号例を以下に示す。まず、符号化の説明で述べたように、符号語のLSB側のゼロラン長によりケースが決まり、ケースが決まった場合、次の符号語の状態は以下の通りとなる。

【0063】

【表10】ケース「0」→状態“0”、“1”

ケース「1」→状態“1”、“2”、“3”

ケース「2」→状態“1”、“3”、“4”、“5”

ケース「3」→状態“3”、“4”、“5”、“6”

【0064】したがって、コードテーブル306の各符号語に対応して出力される復号コードは、上記の取り得る状態にのみ対応して記憶されているので、記憶容量は非常に少なくすることができる。ここで、例えば15ビットの符号語が以下の順番で入力される場合を考える。

【0065】

【表11】

力される符号語に対応して一義的にケースが決まるので、コードテーブル306には、ケースを記憶しておく必要が無く、符号語に対応して、次の符号語の取り得る状態にのみ復号コード語を記憶しておけば良い。したがって、実際に復号する場合には、符号語のケースを算出する必要はなく、入力される符号語を参照アドレスとして次の符号語の状態に対応する復号データをコードテーブル306から読み出すだけで良い。

【0068】

【発明の効果】以上説明したように本発明によれば、入力データ語を符号語に符号化するための符号化テーブルとして入力データ語に対して複数の符号化テーブルを用いるとともに、複数の符号化テーブルが入力データ語に対応する符号語と、次の入力データ語を符号化するための符号化テーブルを選択するための状態情報を有し、さらに所定の入力データ語に対しては、特定の符号化テーブルにおける符号語と他の特定の符号化テーブルにおける符号語をそれぞれNRZI変調した信号が逆極性となるようにしたので、例えば、8ビットのデータをDSV制御を行いながら15ビットの符号語に変換することができ、8ビットデータを16ビット符号に変調するEFM+方式より更にコード化レートを向上させることができる。

【図面の簡単な説明】

【図1】本発明に係る変調装置を備えたディスク記録装置の例を示す概略構成図である。

【図2】本発明に係る変調装置の一実施の形態を示すブロック図である。

【図3】RL L (2, 10) の符号化器の状態遷移を示す説明図である。

【図4】15状態の符号化テーブルを示す説明図である。

【図5】15状態の符号化テーブルから本発明に係る符号化テーブルへの変換例を示す説明図である。

【図6】本発明に係る符号化テーブルの一実施形態を示す説明図である。

【図7】本発明に係る符号化テーブルの一実施形態を示す説明図である。

【図8】本発明に係る符号化テーブルの一実施形態を示す説明図である。

【図9】本発明に係る符号化テーブルの一実施形態を示す説明図である。

【図10】本発明に係る符号化テーブルの一実施形態を示す説明図である。

【図11】本発明に係る符号化テーブルの一実施形態を示す説明図である。

【図12】本発明に係る符号化テーブルの一実施形態を示す説明図である。

【図13】符号化処理を概略的に説明するためのブロック図である。

【図14】図11の変調装置の変調処理を示すフローチャートである。

【図15】本発明に係る復調装置を備えたディスク再生装置の例を示す概略構成図である。

【図16】本発明に係る復調装置の第1の実施形態を示すブロック図である。

【図17】状態演算器の処理内容を示す説明図である。

【図18】取り得る状態番号を示す説明図である。

【図19】取り得る状態番号を示す説明図である。

【図20】図16の復調装置の復調処理を示すフローチャートである。

【図21】本発明に係る復調装置の第2の実施形態を示すブロック図である。

【図22】図21の復調装置が備えるコードテーブル306の例の一部分を示す説明図である。

【符号の説明】

13 8-15変調器(符号化装置、変調装置)

33、33a 15-8復調器(復調装置)

100 符号語選択肢有無検出回路(判定手段)

110 符号化テーブルアドレス演算部(読み出し手段)

120 符号化テーブル(コードテーブル)

130、132 DSV演算メモリ(DSVメモリ)

131、133 バスメモリ

140 絶対値比較部(メモリ制御/符号出力部150と共に選択手段を構成する)

150 メモリ制御/符号出力部

300 同期検出回路

301 シリアル/パラレル変換器

302 状態演算器(算出手段)

303 ゼロラン長検出回路(検出手段)

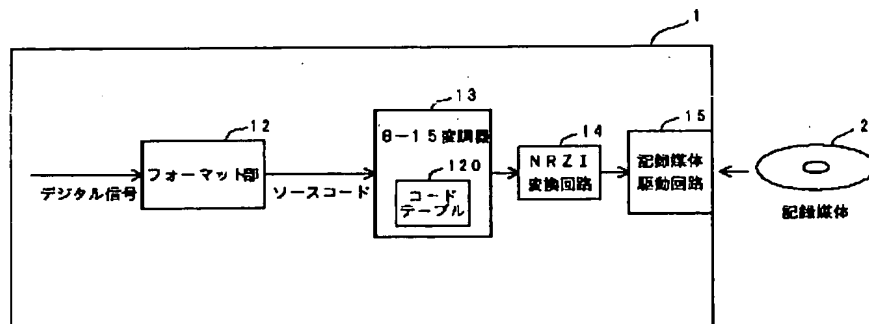
304 ワードレジスタ

305 コード復調部(復調手段)

306 コードテーブル(復号テーブル)

310 状態レジスタ

【図1】



【図 6】

[illegible]

【図 7】

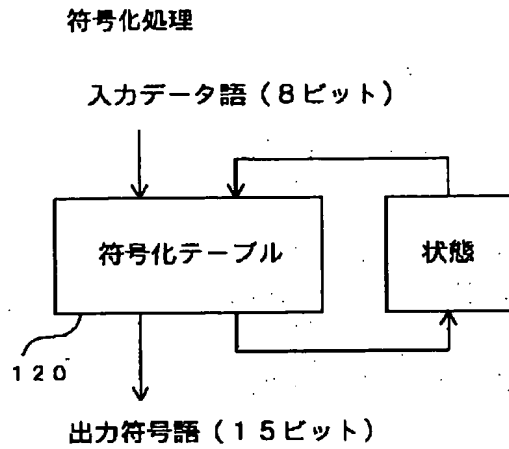
[illegible]

【図 1 2】

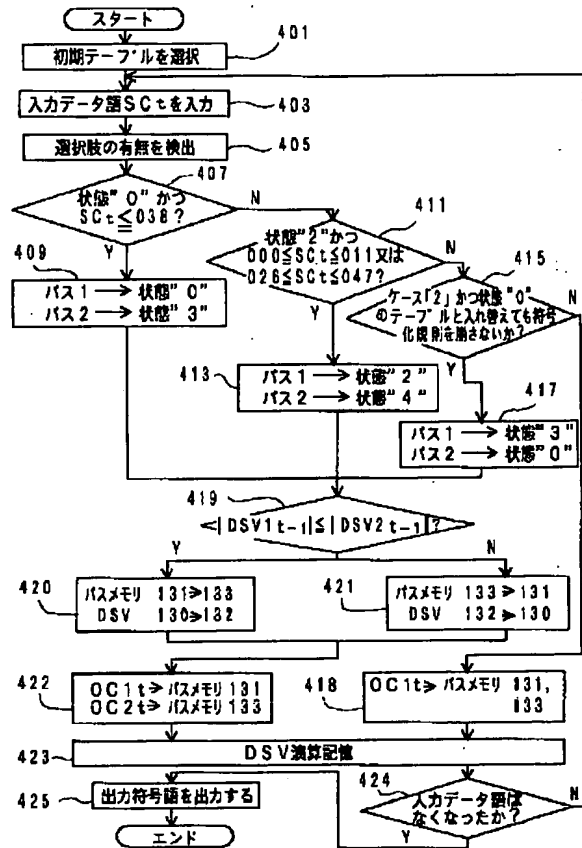
	7 ²			7 ³			7 ⁴			7 ⁵			7 ⁶		
239	001000000000000000	3	001000000000000000	1	000000000000000000	4	001000000000000000	2	010000000000000000	4	100000000000000000	1	010000000000000000	4	
240	000100000000000000	3	001000000000000000	3	000000000000000000	5	001000000000000000	3	010000000000000000	5	100000000000000000	1	010000000000000000	5	
241	000010000000000000	3	001000000000000000	4	000000000000000000	5	001000000000000000	3	010000000000000000	5	100000000000000000	2	010000000000000000	5	
242	000001000000000000	3	001000000000000000	5	000000000000000000	5	001000000000000000	3	010000000000000000	5	100000000000000000	2	010000000000000000	5	
243	000000100000000000	4	001000000000000000	5	000000000000000000	4	001000000000000000	4	010000000000000000	6	100000000000000000	2	010000000000000000	6	
244	000000010000000000	5	001000000000000000	3	000000000000000000	3	000000000000000000	5	010000000000000000	5	100000000000000000	3	010000000000000000	5	
245	000000001000000000	4	001000000000000000	4	000000000000000000	0	000000000000000000	5	010000000000000000	0	100000000000000000	4	010000000000000000	0	
246	000000000100000000	4	001000000000000000	5	000000000000000000	1	000000000000000000	5	010000000000000000	1	100000000000000000	4	010000000000000000	1	
247	000000000010000000	5	001000000000000000	0	000000000000000000	1	000000000000000000	5	010000000000000000	1	100000000000000000	1	010000000000000000	1	
248	000000000001000000	5	001000000000000000	1	000000000000000000	1	000000000000000000	5	010000000000000000	2	100000000000000000	1	010000000000000000	2	
249	000000000000100000	1	001000000000000000	1	000000000000000000	0	000000000000000000	1	010000000000000000	1	100000000000000000	1	010000000000000000	1	
250	000000000000010000	1	001000000000000000	2	000000000000000000	0	000000000000000000	1	010000000000000000	0	100000000000000000	2	010000000000000000	0	
251	000000000000001000	1	001000000000000000	3	000000000000000000	0	000000000000000000	1	010000000000000000	0	100000000000000000	3	010000000000000000	0	
252	000000000000000100	1	001000000000000000	3	000000000000000000	1	000000000000000000	1	010000000000000000	3	100000000000000000	3	010000000000000000	3	
253	000000000000000010	1	001000000000000000	3	000000000000000000	2	000000000000000000	1	010000000000000000	3	100000000000000000	3	010000000000000000	3	
254	000000000000000001	1	001000000000000000	4	000000000000000000	4	000000000000000000	4	010000000000000000	4	100000000000000000	4	010000000000000000	4	
255	000000000000000000	5	001000000000000000	5	000000000000000000	5	000000000000000000	5	010000000000000000	5	100000000000000000	5	010000000000000000	5	

※前のデータ間のゼロラン量が6以下の時	0000100000000000	5
7以上の時	0100100000000000	5

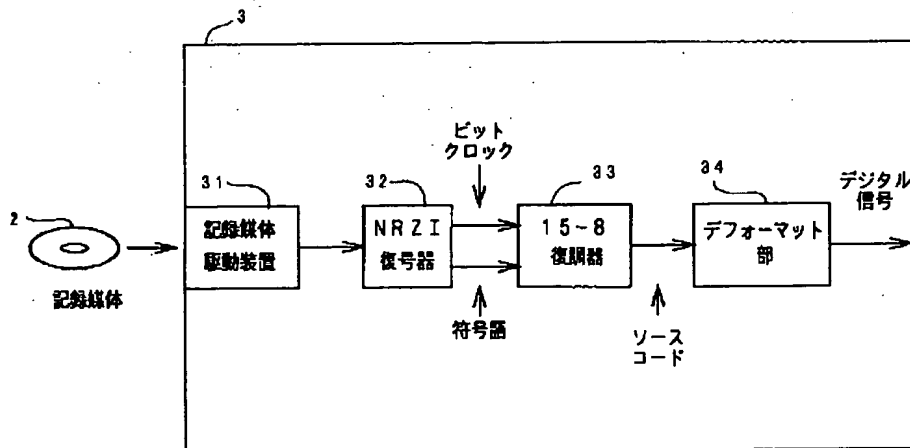
【図13】



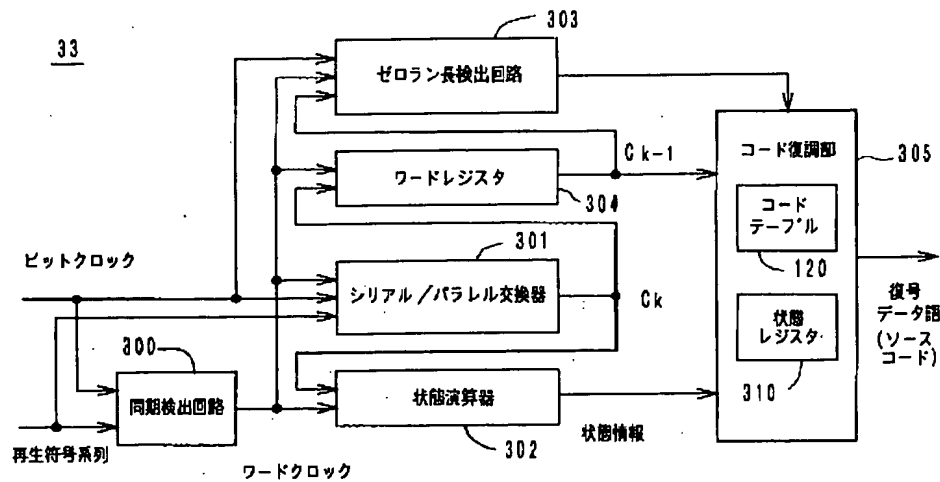
【図14】



【図15】



【図 16】



【図 17】

状態番号

C言語で表現された演算式

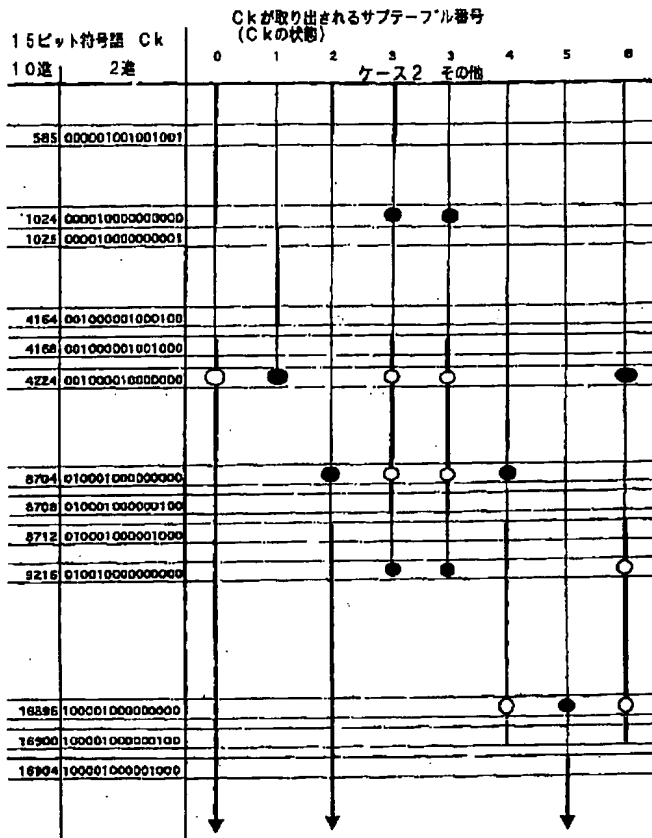
```

0 : (Ck <= 1024) || (Ck) = 4168) && (Ck != 4224)
1 : (1025 <= Ck <= 4164) || (Ck == 4224)
2 : (Ck <= 585) || (Ck) = 8712) || (Ck == 8704)
3 : if (Case == 2) {
      (Ck <= 585) || (Ck == 1024) || (Ck == 9216) || (4168 <=
      Ck <= 8708) && (Ck != 4224) && (Ck != 8704) }
  else {
      (Ck == 1024) || (Ck == 9216) || (4168 <= Ck <= 8708)
      && (Ck != 4224) && (Ck != 8704) }
4 : (Ck == 8704) || (8712 <= Ck <= 16900) && (Ck != 16896)
5 : (Ck == 16896) || (Ck) = 16904)
6 : (Ck == 4224) || (8712 <= Ck <= 16900) && (Ck != 16896)
      && (Ck != 9216)

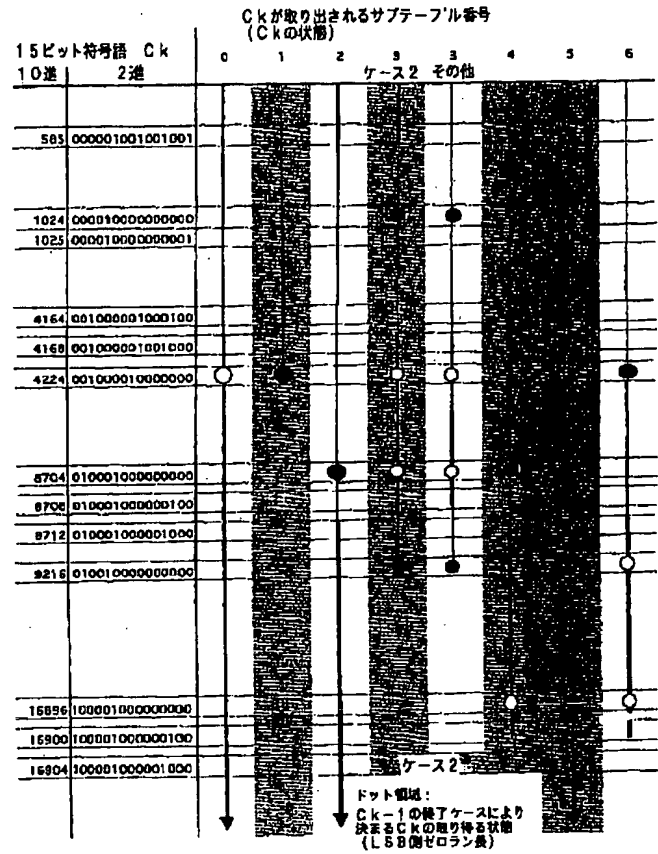
```

(CkはCk-1の次の符号語)

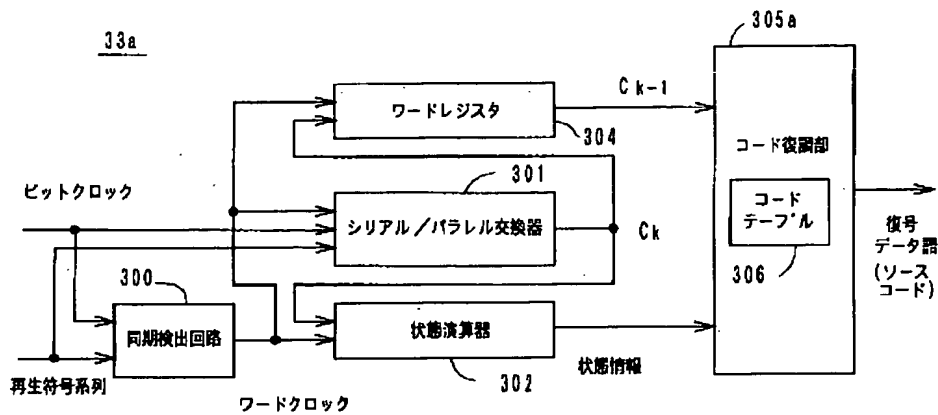
【図 18】



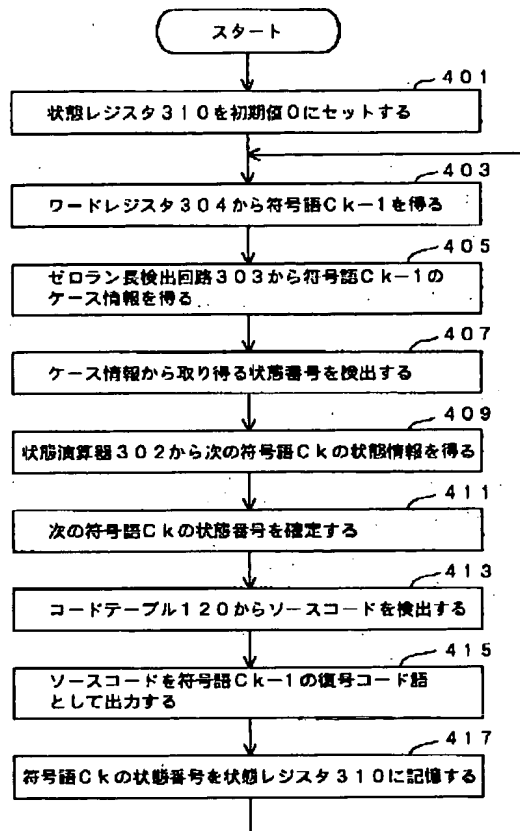
【図 19】



【図 21】



【図20】



【図22】

15ビット符号語 Ck-1			後続符号語の状態番号と復号データ語							
10進	バイナリ (2進)	ケース	0	1	2	3	4	5	6	
16	000000000010000	2	—	157	—	158	159	160	—	
17	000000000010001	0	161	162	—	—	—	—	—	
18	000000000010010	1	—	163	164	165	—	—	—	
32	000000000100000	2	—	—	—	—	0	1	—	
33	000000000100001	0	2	3	—	—	—	—	—	
34	000000000100010	1	—	59	60	61	—	—	—	
36	000000000100100	2	—	4	—	5	6	7	—	
64	000000001000000	2	—	8	—	9	10	11	—	
65	000000001000001	0	39	40	—	—	—	—	—	
66	000000001000010	1	—	41	42	43	—	—	—	
68	000000001000100	2	—	62	—	63	64	65	—	
8704	0100010000000000	3	—	—	—	—	0	1	—	
16904	1000100000001000	2	—	2	—	3	4	5	—	
18722	100100100100010	1	—	249	250	251	—	—	—	
18724	100100100100100	2	—	252	—	253	254	255	—	